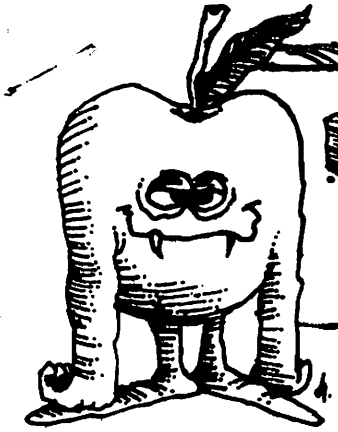


NORTHWEST SUBURBAN APPLE USERS GROUP

THE HARVEST

VOL. 2, NO. 3 NOVEMBER 1980

FEATURE ARTICLE

FLOAT, FLOAT, FLOAT YOUR POINT
(F.P. REPRESENTATION)

by Guy A. Lyle

"Why can't I store a number much greater than ten to the 38th power?" "What is 'floating point'?" "How are floating point numbers stored inside the computer?" "What's the difference between floating point and integer numbers?" "When should I use floating point? Integer?"

I have heard these questions numerous times. Their central theme is the mysterious 'floating point' type of value. So, let us take a look into this mystery to see if we cannot answer some of these questions.

What is Floating Point?

The primary difference between F.P. numbers and integer numbers is that F.P. numbers may have decimal fractions as part of their values. Numbers such as 23, -100, 0 , 6, and 12345 are all integer numbers. F.P. numbers can include such values as 12.1, -16.5, 0.023, -100, and 5000.123. These all possess fraction parts to their values. Note that the value -100 was included in both lists. It may be considered a F.P. value whose fractional part is ".0000....". Therefore, any integer value may also be considered to be a F.P. value, if so desired. The reverse is not always true.

How are they stored differently

Computers store integer and F.P. values differently. Integer variables are stored by the APPLE II's APPLESOFT Basic in two bytes of memory. The coding technique used is known as "2's compliment" notation. Values in the range of -32768 through +32767 are legal using this notation. For some reason, APPLESOFT will not allow the -32768 value, making the lowest value -32767. Integer type variables are denoted by a percent sign (%) following the name of the variable, as in I%.

APPLESOFT stores F.P. values in five bytes of memory. The coding technique used is one variation of the floating point notations generally used. The first byte of the value represents what is known as the "exponent" of the value. The remaining four are known as the "mantissa". Very crudely, the coded value represents a number using the following formula:

value = mantissa * 2 ↑ exponent

CONTINUED ON PAGE 2

Remember that the exponentiation is always performed first. I say 'crudely' because there are some rules regarding the storage of the mantissa and the exponent in APPLESOFT's storage method. Floating point (or 'real') variables have no percent sign or any other sign following their names.

The Technical Details

The exponent byte is stored as an unsigned value, from 0 through 255. It is encoded in "excess 128" notation. That is, the stored exponent is 128 higher than the actual exponent which it represents. Therefore:

actual exponent = exponent byte - 128

This allows the exponent a range of -128 through + 127.

The mantissa is stored to 32 bits (four bytes) of precision. This will allow storage of values up to about 4.2 billion without loss of accuracy. The mantissa is stored as a fraction rather than as an integer value, however. It is assumed that the decimal point is immediately to the left of the leftmost bit. Furthermore, the mantissa is always adjusted so that the leftmost bit is a "1". The legal range of values for the mantissa is ".10000000000000000000000000000000" through ".11111111111111111111111111111111" in binary. These represent decimal values of "0.5" and approximately "0.9999999" accordingly.

The process of forcing the leftmost bit of the mantissa to be a "1" bit is called "normalization". If the result of any computation is not already normalized, then it must be normalized before future use. This process involves shifting the bits in the mantissa to the left to bring the first non-zero bit into position. In order to compensate for the changing mantissa, the exponent must be decremented by one for each position which the mantissa is shifted left. This serves to preserve the value which the entire F.P. number represents. Conversely, any operation which requires that the bits of the mantissa be shifted to the right also requires that the exponent be incremented by one for each position shifted.

The mantissa must also have at least one bit which represents its sign; positive or negative. Since the leftmost bit of a properly normalized mantissa is always a one, APPLESOFT hides this bit and in its place a sign bit is provided. A zero bit represents a positive mantissa while a one bit represents a negative mantissa Therefore the mantissa, in its final form, consists of a sign bit followed by the 31 least significant bits of the mantissa's value. The most significant bit of the mantissa is a "hidden bit" whose value is always "1".

I should add a note here regarding the sign of the F.P. value. It is the sign of the mantissa which determines the sign of the F.P. value. The sign of the exponent merely determines the magnitude of the F.P. value -- 2↑exponent will always be positive regardless of whether 'exponent' is positive or negative. 2↑3 is +8. 2↑-3 is +1/8 or +0.125. This point has always been one of confusion for students when studying the similar topic of scientific notation.

The following table demonstrates the storeage of some common F.P. values. These were produced using a "F.P. VARIABLE PEEKER" program provided at the end of this article.

value (decimal)	F.P. representation (hex)
6	$83 40 00 00 00
3	82 40 00 00 00
1.5	81 40 00 00 00
-1.5	81 C0 00 00 00
0.1	7E 4C CC CC CD

CONTINUED ON PAGE 3

Examining the representation for "6", we see an exponent of $83, or 131 decimal. The actual exponent is 131-128 or just 3. The mantissa shown is .x100000.... The 'x' is the hidden bit, and is always "1". Therefore, the real mantissa is .1100000, or 0.75 decimal. The F.P. value represented is therefore:

$$0.75 * 2 \uparrow 3$$
$$= 0.75 * 8$$
$$= 6$$

Note that the representation for "3" simply has an exponent one less than that for "6". An the same for "1.5". Their mantissas are identical, only their exponents differ.

The "-1.5" value demonstrates the setting of the sign bit in the mantissa for the negative value. Note that the only difference between "1.5" and "-1.5" is the setting of this sign bit. This is true for any positive-negative pair. To find the absolute (positive) value of a F.P. number, simply clear its sign bit. (For you Assembly programmers, that is simply an 'AND #$7F' with the high mantissa byte).

F.P. Operations

The full range of normal operations can be performed on floating point values; addition, subtraction, multiplication, and division. Special rules must be used, however. Those readers who are familiar with the rules of operations when using scientific notation (also known as powers-of-ten notation) have a decided advantage here. The same rules apply.

Multiplication and division are the easiest to work with. There are three steps involved: (1) The "actual" exponents are added (for multiplication) or subtracted (for division). (2) The mantissas are multiplied. (3) The result is then properly normalized. Overflow will result if the actual exponent exeeds 127. Underflow occurs if the exponent is less than -128. APPLESOFT generally ignores underflows, turning the result to "0". Incidentally, "0" is a special case -- all five bytes are $00 in representing "0".

Truncation of accuracy is easily seen when looking at multiplication. When two 32-bit values are mutliplied, the result is a 64-bit value. Only the most significant (left-most) 32 bits of this result are maintained in the final result. The other 32 bits are generally just chopped off. Sometimes a rounding-off algorithm may be applied, although I do not know whether or not Applesoft applies such an algorithm. If there are any non-zero bits in the 32-bit chopped off portion, then loss of accuracy results. The 32 bits of maintained result represent about 9 1/2 digits of decimal accuracy (1 in 4.295 billion).

Addition and subtraction generally involve a more complex process. Before these operations can take place, both exponents must be made equal. This is generally done by pre-adjusting the mantissa and exponent of the value having the least exponent. For each position the bits of the mantissa are shifted right, the exponent must be increased by one. This process continues until both exponents are equal. At this time the mantissas can be added (or subtracted). At times 32-bit additions can produce 33-bit results. Some subtractions can produce less then 32-bit results. In either case, normalization of the final result must take place.

Addition or subtraction of value with very large differences in magnitude can produce unusual results. This is due to the pre-adjustement of the least-exponent mantissa prior to the operation. If the difference in the exponents is greater than 31, then the mantissa of the least-exponent number would be complete shifted away, leaving a mantissa of "0"!! Adding (or subtracting) two such wildly different numbers would produce a result equal to the larger number alone.

F.P. vs Integer

So what's the advantage of F.P. numbers? The primary advantage is the ability to express fractional portions conveniently. There are many things in the world which simply cannot be

CONTINUED ON PAGE 4

expressed well as integer values. The nine-digit accuracy of APPLESOFT's F.P. values also offers more utility to the user, as opposed to the 4 1/2 digit range of integer values. Many other BASICs only offer six or seven digit accuracy in their F.P. values. This is because they only offer a 3-byte mantissa. One could argue that double-precision (say four or five byte) integer values could have solved this problem, but this could not provide for fractional values nor the large range of exponents available in F.P. values.

F.P. values are not without their problems, however. One major problem is the time which it takes to perform F.P. operations. Much more bit-twiddling is involved in F.P. operations as compared to straight, everyday integer operations. Programs using F.P. operations are significantly slower than those involving integer operations. This is one of the primary reasons that APPLE II Integer Basic programs execute so much more rapidly than APPLESOFT Basic programs do. This also brings up one of the basic failings of APPLESOFT Basic -- It fails to take advantage of easier and quicker integer value operations. In an expression containing Integer type values and variables, all values are converted first to F.P. values before operations with them are effected! (See APPLESOFT Reference Manual, page 18, second paragraph from bottom.) This process of integer-to-F.P. conversion adds even more time to evaluating the expression. It is fairly easy to demonstrate, by timing repeated execution loops, that such expressions take longer to evaluate. FOR-NEXT loop execution alone would be significantly increased if the authors had allowed the use of integer-type variables for the control variable. I have used a Basic which does allow this and the difference can be quite noticable.

The question becomes one of "Why bother with integer-types?". In terms of APPLESOFT Basic, there are several things which can be stated in favor of integer-types.

Some Basic functions expect integer-type arguments. When such functions receive F.P. arguments, time must be taken to convert them to integer values, while integer values would be used directly.

I have also made use of the integer-type variable to denote certain types of values within programs. Memory addresses, for example, may all be maintained in integer variables. As such, these represent more of a programmer's aid and do not offer much of any aid to Basic.

The most signficant area of usage for integer variables is in large numeric arrays. A 100x100 array of F.P. values would require about 50,000 bytes of memory -- impossible within the APPLE II. If an integer array were used, only about 20,000 bytes of memory would be needed -- possible, depending upon the size of the program. This is a savings of 60%!. The only restriction is that the values to be stored must be representable in integer format: -32767

through +32767.

It should be pointed out, however, that there is no space savings in using simple (non-array) integer variables. Each numeric simple variable, F.P. or integer, consumes seven bytes of memory. For integer variables, three of the seven bytes are not used.

FLOATING POINT VARIABLE PROGRAM

A program which shows how floating point values are stored in the computer's memory has been provided. (Hopefully the editor(s) will not forget to print it along with this article.) The user RUNs the program and enters any F.P. value when requested. The program then prints out the hexadecimal representation for that value.

The representation of the value is taken directly from the table which Applesoft keeps of all its simple variables. The variable V is used before any other variables in the program in line 150. This insures that it will be the first entry in the table. Line 160 computes the memory location of the first of the five bytes which will hold V's value.

CONTINUED ON PAGE 5

The FOR..NEXT loop from line 210 through line 270 scans through these five bytes, PEEKing their values from memory and displaying them on the screen. The subroutine at line 300 is called twice within the loop, printing one hexadecimal digit each time (in line 310). Line 220 computes the high order digit to be printed by the subroutine, line 240 the low order digit. Line 260 simply prints the space between each byte value.

```
]LIST

100  REM   FLOATING POINT
110  REM   VARIABLE FORMAT
120  REM
130  REM   BY GUY A. LYLE
140  REM
150 V = 0: REM  1ST DEFINED
160 LOC =  PEEK (105) + 256 *  PEEK (106) + 2
170  HOME
180  INPUT "ENTER A VALUE: ";V
190  PRINT
200  HTAB 5: PRINT "$";
210  FOR I = LOC TO LOC + 4
220 :NYBBLE =  INT ( PEEK (I) / 16)
230 : GOSUB 300
240 :NYBBLE =  PEEK (I) - 16 * NYBBLE
250 : GOSUB 300
260 : PRINT " ";
270  NEXT I
280  PRINT : PRINT
290  GOTO 180
300  REM  ** PRINT HEX CHARACTER **
310  PRINT  CHR$ (48 + NYBBLE + 7 * (NYBBLE > 9));
320  RETURN
```

HOUSE CLEANING

Skip Neiburger DDS
(c) 1980 Andent Inc.
Reprint permitted for users
groups.

(At the last club meeting, a member asked me if there was a way to clean the disk drive without spending the $20-30 on a diskette cleaning kit. Coincidently, Skip had sent this as one of several articles for the newsletter. I am cautious of Skip's approach here in regard to any unqualified individual tackling this problem as described below. However, I see no harm if you know what you are doing. DMA.)

Some day, if it hasn't arrived already, you may experience some odd behavior involving your floppy disk drives. You may notice incomplete program loads, lost data or occasional failures to boot-up (load a disk to the computer). It may happen only at odd times or occur with great regularity - being inconvenient at best, which tends to cause some concern. It is of little problem if you are playing a game, but what disasters can compare with this 'glitch' occuring in the middle of billing or when your lawyer asks for a copy of your patient's record for a malpractice defense. That's enough reason to DEcomputerize.

Many of these problems are due to improper maintainence of your disk drives and diskettes. The biggest culprit is dust. Small specks will settle on your disk and eventually be carried into the disk drive where the sensitive read/write head resides. The read/write head contacts the disk surface and senses or alters the magnetic iron oxide (data) on the disk. If dust gets in the way, data may be lost, especially in older, well-used disks in which the iron oxide has been worn thin (less magnetic field).

One can correct much of this problem by cleaning the head and head pads with a soft, dry paintbrush. Remove the drive cover and blow out the dust. (Use a rubber ball with a pipe on it that blows air when squeezed, NOT the canned air sold in photographic stores. Sometimes, the freon propellant escapes with the air. ED.) Wisk off the dust clinging to the head mechanism with the brush. Then check each socket assembly by plugging and then

CONTINUED ON PAGE 0A

CLUB NEWS

CLUB OFFICERS

President: Don Fuller	312-991-6868
Vice Pres: Mike Robins	312-593-2709
Secretary: Joel Kurasch	312-677-8358
Treasurer: James Wilson	312-358-2196
Librarian: Mike Rose	312-359-4306
Asst. Lib: Julian Vassay	
Prog. Crm: Ken Rose	
NL Staff: Jim Nowak	

Club Addresses:

MEMBERSHIP, etc--BOX 787
Palatine, Il. 60067

NEWSLETTER-------Dave Alpert
880 Mellody
Lake Forest, Il. 60045
312-295-6078
SOURCE#-TCA 640
MODEM 312-295-6926

BULLETIN BOARD---312-991-6245 daily
as available and after
10:30pm local time

Membership is open to all. Dues are $12.00 annually. New members are required to supply one disk or cash equivalent ($5.00) at the time of admission to the club. Disk not required of renewals. Membership applications are available from the club Secretary at the meetings or by mail.

NSAUG NEWSLETTER is published by the Northwest Suburban APPLE Users Group and edited by Dave Alpert. Contents may be reproduced by other APPLE clubs for their newsletters or bulletines except where specifically reserved by the author. Permission to copy these articles may be obtained by writing the author through our editorial offices. Please indicate THE HARVEST as the source of any material reproduced. Newsletter exchange is welcomed. The Harvest recognizes that APPLE and APPLE manufactured products are trademarks of the APPLE Computer Company.

MEETING SCHEDULE

Regular Meeting Sat. November 1
Sat. December 6
Sat. January 3*
*tentative

Officers meetings are held at the President's home on the Tuesday evening following the regularly scheduled meeting starting at 7pm.

Regular meetings are held at the Schaumberg Library, 100 Library Lane, Schaumberg, Il.

MEMBERS AIDE

The members listed below have volunteered to answer questions from club members who need a 'HOT LINE' type approach. Please try to be brief when you call as a courtesy to them.

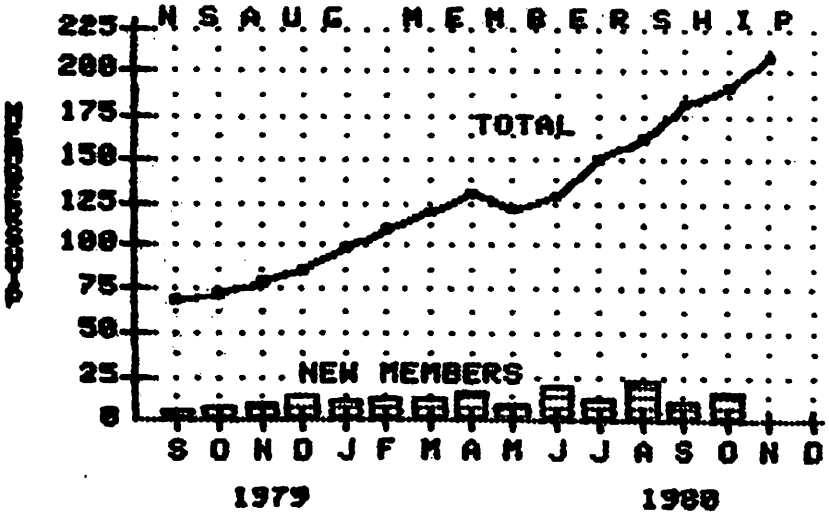
Listed below are the names and phone numbers. The numbers after their names represent their special talent. PLEASE-no calls after 10pm.

Earl Allen	312-837-9259	1-9
Mike Robins	312-593-2709	9-0
Paul Sladfield	312-359-2378	4-9
Guy Lyle	312-438-7941	4-9
Ted Roseman		
Mary Roseman	312-338-4833	2,3
Joel Kurasch	312-677-8358	9
Leon Alexander	312-725-5309	9
Gordon Banks	312-324-6194	0
Rich Lundeen	312-420-2008	2,3

Mach. Lang=1	Int. Basic=2
Applesoft =3	Hardware =4
Arrays =5	DOS =6

All of above=9
PASCSAL=0 (not included in #9 designation)

When numbers appear befor a 9, the person is especially qualified in that area as well as in all other areas.



MORE CLUB NEWS

SECRETARY'S REPORT

Regular Meeting
Meeting of October 4, 1980
ATTENDANCE: 130

The meeting was called to order by Program Chairman Ken Rose. Librarian Mike Rose talked briefly about the library. Treasurer Jim Wilson reported on the club finances (see Treasurer's report). Ken Rose then reminded everyone about the contest, the deadline for which is December 31. Dave Alpert, the Harvest editor, called for questions for the Ask Mr. Apple column and other material for the newsletter. Vice-President Mike Robins spoke about the I.A.C.

Ken Rose then gave a presentation on Exec files. It was shown how these files could be used to do many things, including creating other programs.

Dave Alpert then demonstrated how to draw a dog using the text screen. In the presentation, Dave explained how memory is mapped for the crt display.

Paul Stadfield then presented a drawing, in high resolution graphics, with the use of a "sky writer". Using a method of vectors, as opposed to HPLOTS, he showed the independance a programmer can achieve for locating a display on the screen.

There was then an update on games available. Among the games discussed were: Dog Fight, Flight Simulator, The Wizard and the Princess, and Stellar Trek.

Mr. Apple then informed the membership that the National Computer Conference is looking for presenters for it's show next May, in Chicago. The Apple /// is facing delays in production, and it is suspected that they will not be seen in the stores till '81. DOS 3.3 is being re-released, with the known problems resolved. The future offering of Apple Computer stock was also discussed.

There were 17 new members enrolled at the meeting, bringing the count to 208. In addition there were 24 renewals.

Officers Meeting
October 7,1980

Program chairman Ken Rose discussed the December agenda which was approved by the officers.

Reports were made by the following committees:

1. Dinner dance
2. Facilities

3. Nominating

The secretary, treasurer, librarian, newsletter editor and IAC representative made reports. Major topics discussed were disk purchases for the club librarian, the new constitution, and a look at the prospects of incorporation.

Respectfully submitted,
Joel Kurasch

```
***************************************
*                                     *
*              NOVEMBER               *
*               AGENDA                *
*                                     *
*                                     *
* 10:00-10:30 OPENING REMARKS         *
*             Ken Rose                *
* 10:30-11:30 PRINTER TECHNOLOGY      *
*             Terry Tufts             *
* 11:30-11:45 BREAK                   *
*                                     *
* 11:45-12:15 PROGRAM LINE EDITOR     *
*             DEMONSTRATION           *
*             Dave Alpert             *
* 12:15-12:30 REPORT ON S.I.G.S.      *
*             SIG Leaders             *
* 12:30-12:55 ASK MR. APPLE           *
*             Mike Robins             *
* 12:55- 1:00 PRESIDENT'S REMARKS     *
*             Don Fuller              *
*                                     *
***************************************
```

TREASURERS REPORT

Month ending Sept. 31

August Ending Bal.	**$941.27**
Membership	**146.00**
Newsletter	**<139.30>**
Disk Project	**177.50**
Misc. Income	**38.06**

Sept. Ending Bal.	**$1163.53**

Respectfully submitted,
Jim Wilson, Treasurer
Oct 8,1980

PREZ SEZ!

November already, and who knows how deep the snow will get.

Renewals! Got yours in? So far, we have 24 renewals -plus- 18 new members so the count for 1981 is 42.

Just received a copy of 'The Orchard'. If you don't have a copy and are seriously into programming, run, don't walk to your favorite computer store for a copy. There is a subscription form in the magazine for individual subscriptions. If you have a copy of 'The Orchard' and are wondering wha' hoppen to 'NSAUG's address, that just happens to be the vice presidents abode. No problem, we'll still get the mail and hopefully, the next issue will contain a correction.

Marty Rutstein will be filling in the details of the upcoming social event next February. The entertainment he has lined up sounds tremendous. The club will be subsidizing some of the cost to keep your cost at a reasonable amount. Come one, come all...

October's meeting... if you are interested in a followup on 'exec' files, see some of the previous issues of 'The Harvest' (Subroutine Sam's column). Now that we know how good Dave Alpert and Paul Stadfeld are at 'walking the dog', I expect to see some of their work appearing in 'The Harvest'... if you weren't there to get the handouts, toooo bad.. Pauls use of data statements to get Snoopy on the high res page leads to all kinds of possibilities.

I trust that by this time you have received a copy of the proposed new constitution. If you haven't, you will. The cover letter explains the reasons I think we need a new constitution and why we need it at this time.

The library is now set up to mailorder disks. We would like to ask that you order disks at the meeting if at all possible. The time and effort spent on mail orders could quickly get out of hand and the librarian may get a bit testy if you overload him. Again, this was set up primarily for those who live quite a distance away and can't come to the meetings regularly. I might add that this club is operating on volunteer services and if the workload gets too big, we may have a volunteer problem.

Speaking of volunteers. If you want to do something for the club, just let me know (so there is no redundancy) and we'll try to get you in motion on your idea. Constructive efforts by members are encouraged at all times for the betterment of NSAUG. A good place to start is the suggestion box at the meetings or a note in the mail.

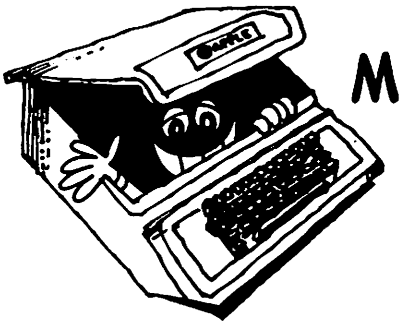
Don

CONTEST NEWS

The deadline for our program contest is December 31, 1980 and will provide an opportunity to win some great prizes for the entrants.
Larry Erdman, Assistant Contest Chairman has contacted all Computer Stores in the Chicago area and thus far reports that nine of the fourteen have either pledged or donated additional prizes. The first and second prize from NSAUG is $100 and $50 respectively. Prizes from our friends in the Computer business range from gift certificates of up to $50, software and a black and white monitor. A complete list of donors and their gifts will be listed next month.
The entries will be judged on originality, cleverness and programming efficiency. They will become part of the club library, available to all members, with commercial rights reserved for the author.
Winners will be announced at the club banquet in February and a complete list will be published in the March Harvest.
Ken Rose, Contest Chairman stated there will be special categories for novice and advanced programmers and there should be no hesitation in entering any program. His own program, Print Hello (5 Print Hello, 10 End), is currently one of the leading contenders.
Programs may be submitted to Ken at the November meeting...an Apple will be set up for that purpose. He may also be contacted at 359-4306 to set up mail or other means of receiving the programs.
Get to work and get those programs in.

FAULTY DISK?

If your library disk shows an "I/O" error when trying to boot, bring it back to Mike Rose for a trade in. It's a faulty disk.



ASK MR. APPLE

Mr. Apple,

As a member of the club I have looked over many of the disks in the club library. Upon running some of these programs, I noticed that some of these programs will crash into the monitor "*" and others might just hang for no reason, and the only way that I can get control back is to hit reset. I have also noticed that a couple of the integer basic programs might give me a "> 32767 ERROR. Do you have any answers to these problems?

Concerned Member

Dear Concerned Member,

There have been some comments about the library in regards to some of the users having difficulty running some of the programs. I hope in this answer to provide some hints about using some of the programs in the library.

First of all let me say that the program library is a contributed library from our members, and members of other groups around the world (we do have a disk from the Apple Users group in Japan) and as such the programs and the documentation for the programs is only as good as the form that we received it from the contributor. We should all be aware that we are getting these programs for nothing and that basically we should not expect that these programs be as well documented as a commercial grade program. The club software committee does make an effort to make sure,to their best ability, that the programs will run, but they do not spend many hours with the programs and there may be bugs that might show up with continued use.

The second thing that I'd like to cover are some hints that might make it possible to run some 'unrunnable' software. Some of the programs in our library require that the user of the program have certain hardware in his machine. An example of this might be the Mountain Hardware clock for a version of 'Oregon Trail' or a special printer to run a graphics demo. There are also some graphics programs that require that the "Programmers Aid #1" be installed. All of these things can cause the novice programmer a lot of headaches. Many times if you look at the program there may be a comment about what special hardware the program needs to run. A clue to the use of a programmers aid chip might be that the program will crash out and go into the monitor at an address around $D800 or something like that (this is the location in memory where the programmers aid software resides. If it's not there when the program calls it , it will crash)

Some of the other errors with programs beset the user with a 48K Apple. One of these that can be encountered is an integer basic program that when run gives a "> 32767 ERROR". Many times the solution for this is to enter the direct command "HIMEM:16384" which will set the pointer for HIMEM to a value that integer basic can calculate. In these cases, the contributor had a smaller size machine when he wrote the program, and had never given thought, that when he does a calculation, that the answer would exceed the numerical valee of "32767".

Another problem can be the user who has a 16K or 32K machine, and he trys to run a program that is too big for his machine. Some of the results can be rather strange. The first and most obvious error is that upon trying to load the program, that it comes up with the error "PROGRAM TOO LARGE". Well that kind of says it all, and should be quite obvious. The other possibility is that the program will load properly, but it may use too much variable space and an error "OUT OF MEMORY" may occur.

Other conflicts may arise on the smaller machines when using the HIRES screen. A user of DOS with 32K of memory does not have

CONTINUED ON PAGE 0A

ASK MR. APPLE

access to the second page of HIRES, due to the fact that DOS is occupying that area. Of course if the program is trying to use that second page, you will find that your program will probably crash with very unpredictable results. The same type of problem also exist with users of Applesoft Basic in RAM., but this time it is slightly different. The first HIRES page is occupied by the Applesoft in RAM and again there can be unpredictable crashes.

The Third problem is a program that uses a loads in a binary file, and crashes. There are some programs that don't include statements to protect the machine language portion of the program from the basic veriable table by either using a "LOMEM:" statement or by using a "HIMEM:" statement. The results of this are a program that might hang or crash into the monitor. A solution for this might be to find out the binary length and starting address of the program and manually setting HIMEM or LOMEM to the appropriate value and then try rerunning the program.

I hope that this has answered some questions about why programs might crash, and it should also be noted that these tricks apply not only to programs in our club library, but to commercial programs as well. Not every programmer writes user-proof software. I also hope that my answer might stimulate some question from the membership, that can be used in future Mr. Apple columns. If you have a question for Mr. Apple, please send it to the NSAUG P.O. Box or submit it at the meeting to any of the officers,or news letter editor.

Mr. Apple

WANT ADS

Dave Drucker wants to sell!
Paper Tiger with graphics option
Decision Data 6540 printer
Applesoft card

His phone number is 312-541-2122

Jim Wilson is interested in hearing from game players. 312-358-2196

Mr. Apple is interested in Getting questions from anyone. P.O. Box 787 Palatine, Il. 60067

HOUSE KEEPING FROM PAGE 5

unplugging each connection. Look for corrosion and polish these areas with the end of a pencil eraser (avoid making more dust with the eraser scraps). Reconnect. Gently press each IC chip (after grounding yourself). This will often improve the electrical contact between the chip and it's socket. Unkink the cables and test for proper function. Occasionally a small wire will work loose due to repeated flexing thus producing intermittant electric contact.

Take good care of your disks too, being careful to protect the surface from dust by inserting them in the envelope immediately after use. Remember the read/write head contacts the UNDERSIDE of the disk so that placing it naked on a table top may cause some friction on the data surface. Also note that most disks are good for only 40 hours of continuous use. A wise person will make back-ups and exchange them frequently with the originals on a regular basis.

By taking these precautions and applying a little preventive medicine, you can help reduce the incidence of disk errors.

****

MEMBERSHIP RENEWALS

Please look at the mailing label on this newsletter. If you have an '80' under your name, then you have not renewed for 81. Renewals for 81 must be in by December in order to continue receiving membership benefits. The mailing list will be expunged of delinquent names about the 5th of December.

Applications were included in last months Harvest. If you have misplaced yours or did not receive one, please call the club secretary. Please note that renewals are $12.00 and do not require that a disk be given to the club. Renewals may be mailed to the club PO Box or dropped off at the next meeting.

We are shooting to equal or exceed last year's renewal rate of 97%. Please attend to this matter as soon as possible.

Thank You,
Your Officers

Copy 1

NSAUG DISK ORDER FORM

Please fill out and give (or mail) top copy to the club librarian with the proper fee. If mail order, enclose check only. NO CASH. Indicate the volume numbers of the disk you want below.

( )

Date ordered________________

Total no. of disks ordered_________ * Librarian's use only *

* $3.50=$__________ Date (ready-mailed)______/______/______

Mail svc.* $1.50/2 disks=$__________ ( )Cash ( ) Amount $__________

Total enclosed $__________

Name(please PRINT)_______________________________________Address____________________________

________________City__________________________________State__________________Zip____________

*** TEAR HERE ***

This copy for your records. Copy 2

Disks ordered:

( )

Date submitted_______________Amount paid_______________

NSAUG'S ST. VALENTINE'S DAY MASSACRE DINNER
$25.00--PER COUPLE
(Single "guys" and "gals" $12.50 each.)
THE BIGGEST EVENT OF THE YEAR

Set aside Saturday night, Feb. 14, 1981 for our first annual dinner. NSAUG'S ST. VALENTINE'S MASSACRE DINNER will be held at the new Deerfield Golf Course Club House.

We promise you a Cantonese buffet dinner from one of the best kitchens this side of Peking (Illinois).

Cash bar--Prohibition style.

At the dinner we will have the announcements of our software contest winners, installation of our 1981 officers, entertainment by an outstanding mentalist and a few Apple surprises.

Incidentally, we'll also have prizes for those individuals and couples whose attire best fits the 1930 gangster and flapper theme. Otherwise, dress informally.

Oh yes, no loaded weapons allowed!

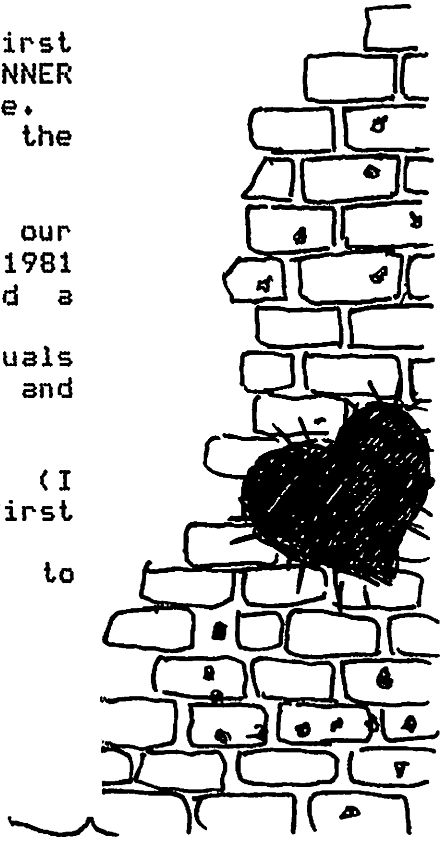
Reservations are a must as we have room in the garage (I mean club house) for only 180 persons. First come, first served.

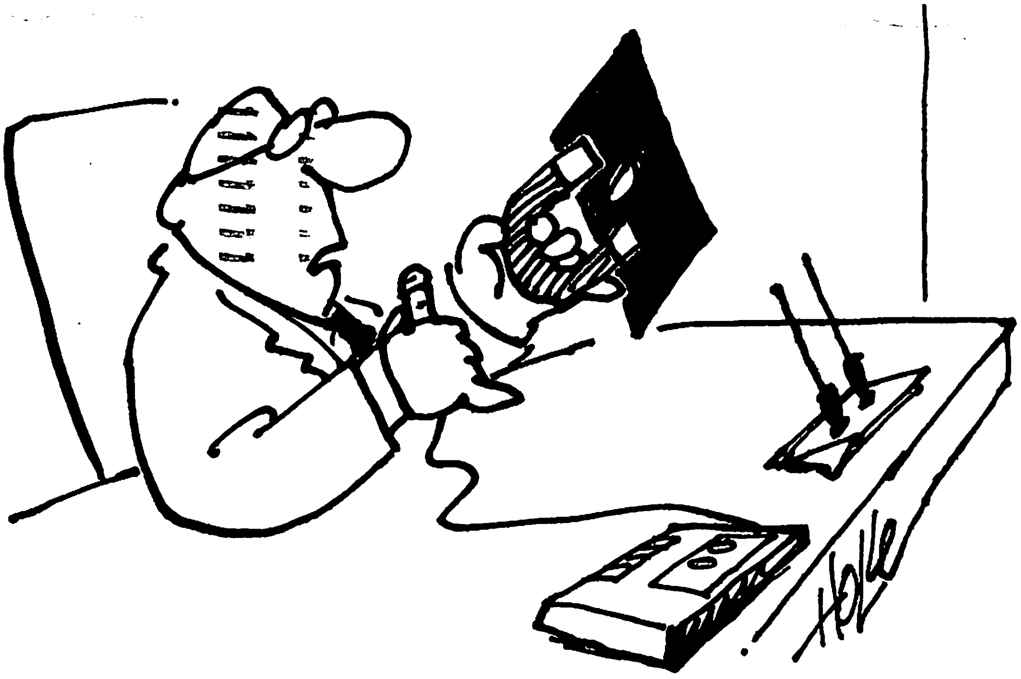
Sorry, but this February 14 slaughter is only open to persons from High School age up.

Mail checks made out to:

NSAUG
c/o Martin Rutstein
480 Juneberry Road
Riverwoods, Il. 60015

(Include your name and telephone number)





" Dear MR. APPLE: CONCERNING MY
PREVIOUS CORRESPONDENCE ABOUT
DISKETTE STORAGE... "